LINKÖPING UNIVERSITY

Department of Electrical Engineering

Linköping University
INSTITUTE OF TECHNOLOGY

IMPLEMENTING THE FFT ON GPUs

TIPS & TRICKS

Mario Garrido Gálvez
mariog@isy.liu.se

Linköping, 2013

MARIO GARRIDO

- Associate Professor at ES, ISY
- PhD in Electrical Engineering (Spain).

- Research background:
  - Optimized implementation of signal processing algorithms.
  - Transforms (FFT, STFT,…), statistical operations (regressions, median filter,…).
  - Data management (matrix transposition, interleavers,…).
  - Hardware designer (FPGAs, ASICs,…).

A STORY ABOUT GPUs

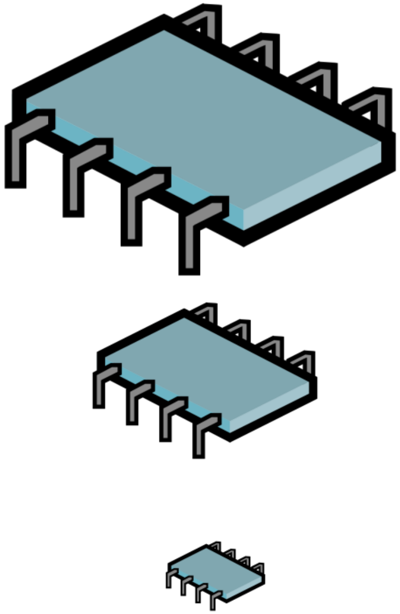
Once upon a time…

< 2011

OPTIMIZE

OPTIMIZE

OPTIMIZE



Mainly FFTs on FPGAs.

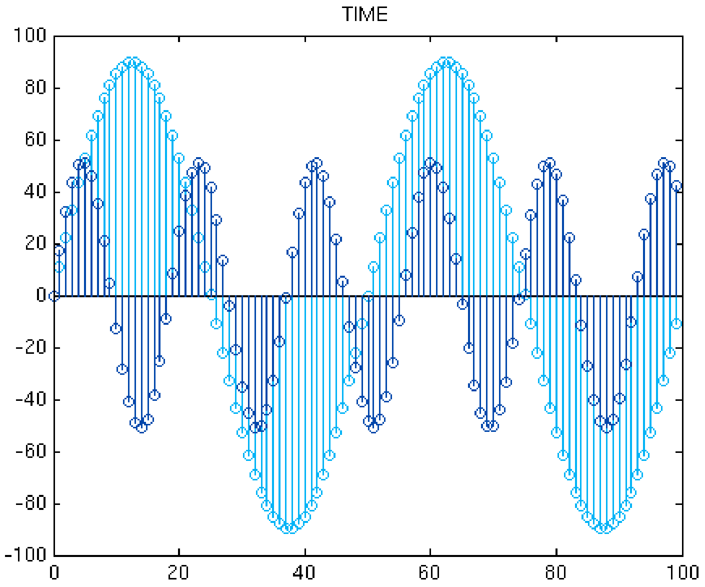
Hundreds of papers in the topic since the 70’s.

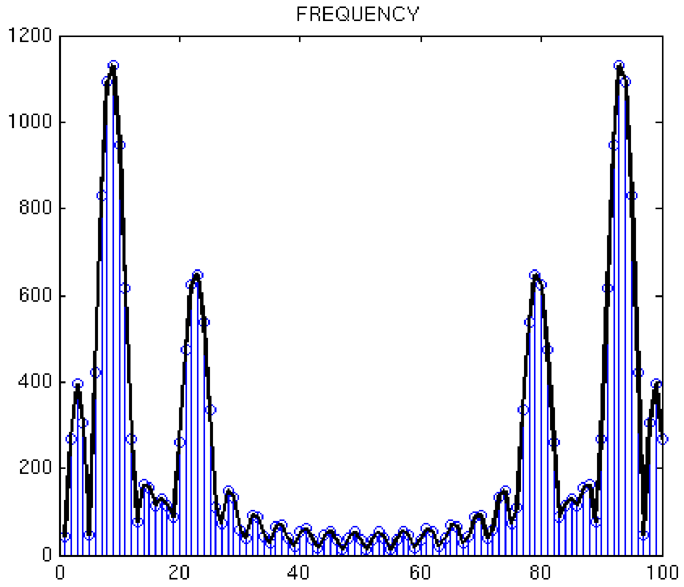
Is not everything done???

DFT / FFT

- Discrete Fourier Transform / Fast Fourier Transform.
- The most widely used algorithm in signal processing

  - Audio and Image Processing. - 3G, 4G.
  - Medical applications: EEG, ECG. - ADSL.





…,2011,…

FFT FPGA FFT FFT FPGA FPGA FFT FPGA

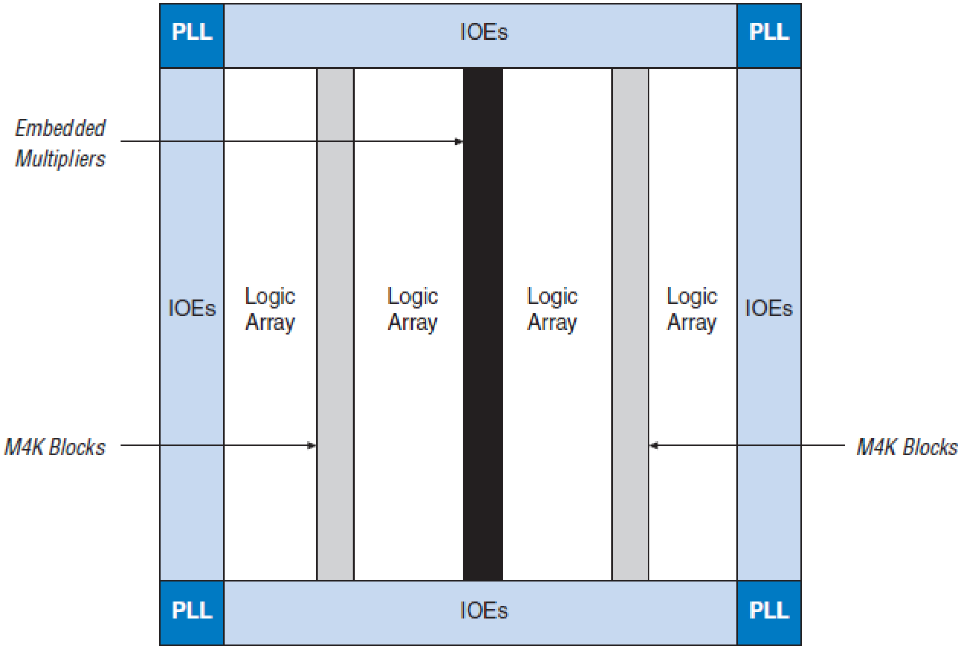
FPGA FFT FPGA FFT FFT FPGA FPGA FFT…

I should do something new!!

What about GPUs?…Shouldn’t it be the same…

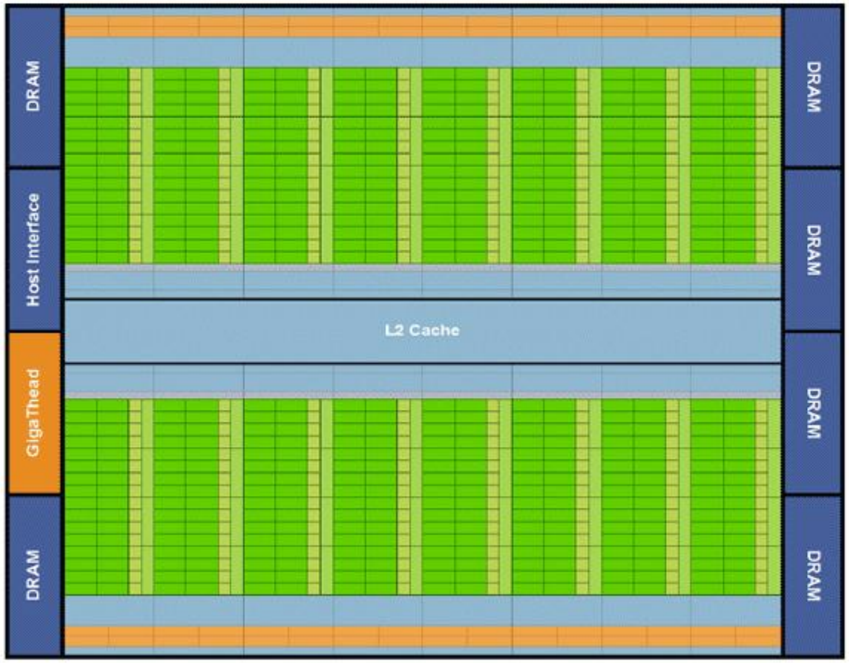
FPGA vs GPU

FPGA



Altera Cyclone II

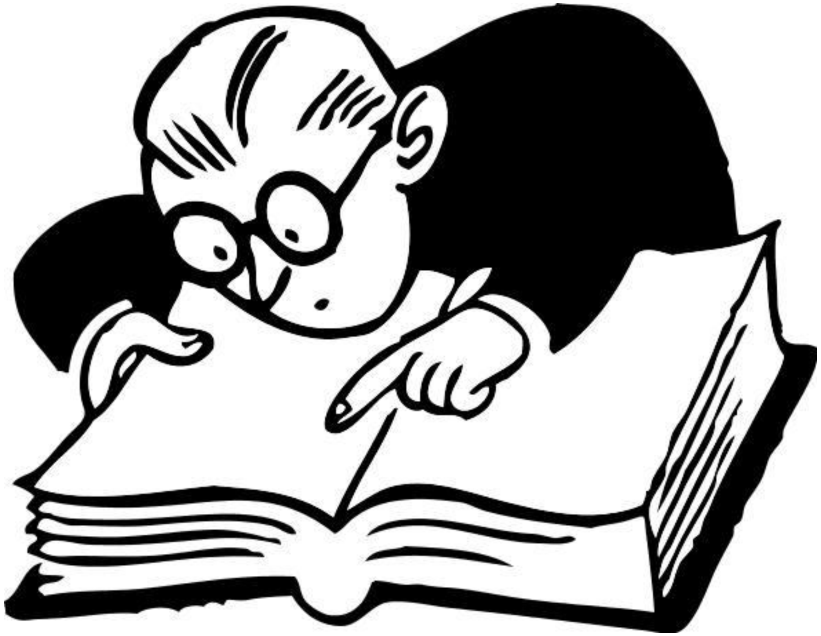
GPU



NVIDIA Fermi

…,2011,…

- Started Master Thesis (Sreehari Ambuluri): FFTs on GPUs.
- Read articles and a book on GPUs.
- Asked Ingemar, Jens, Gabriel.



…,2012,…

Finish the Master Thesis.

The work is good.

Why not to improve it and publish a paper?

Asked Ingemar, Jens and Gabriel for collaboration.

…,2013,…

5x

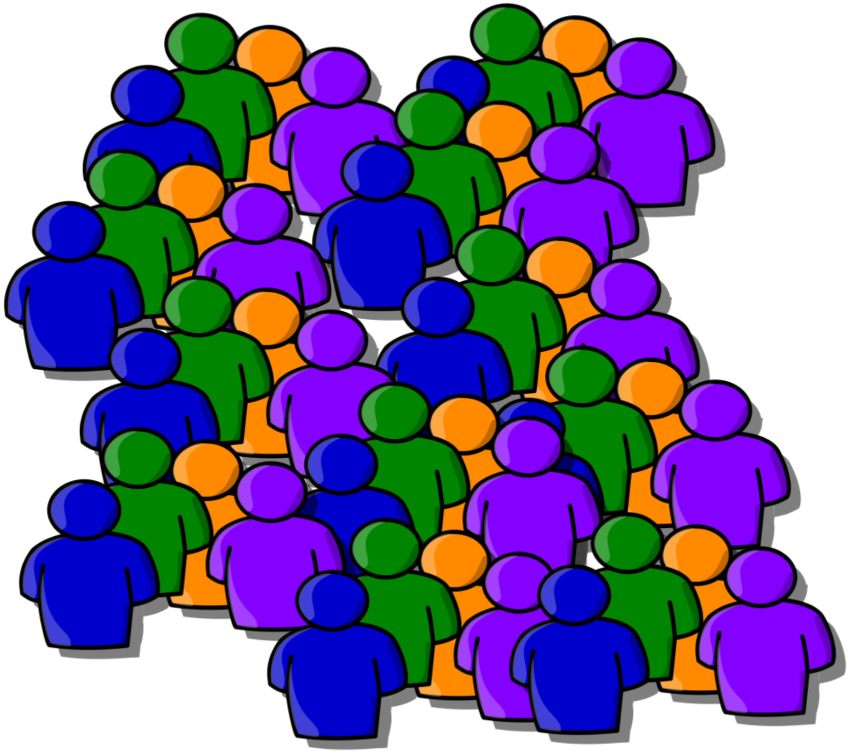
MCCSIS

…,2013

BEST PAPER AWARD

NVIDIA

NVIDIA

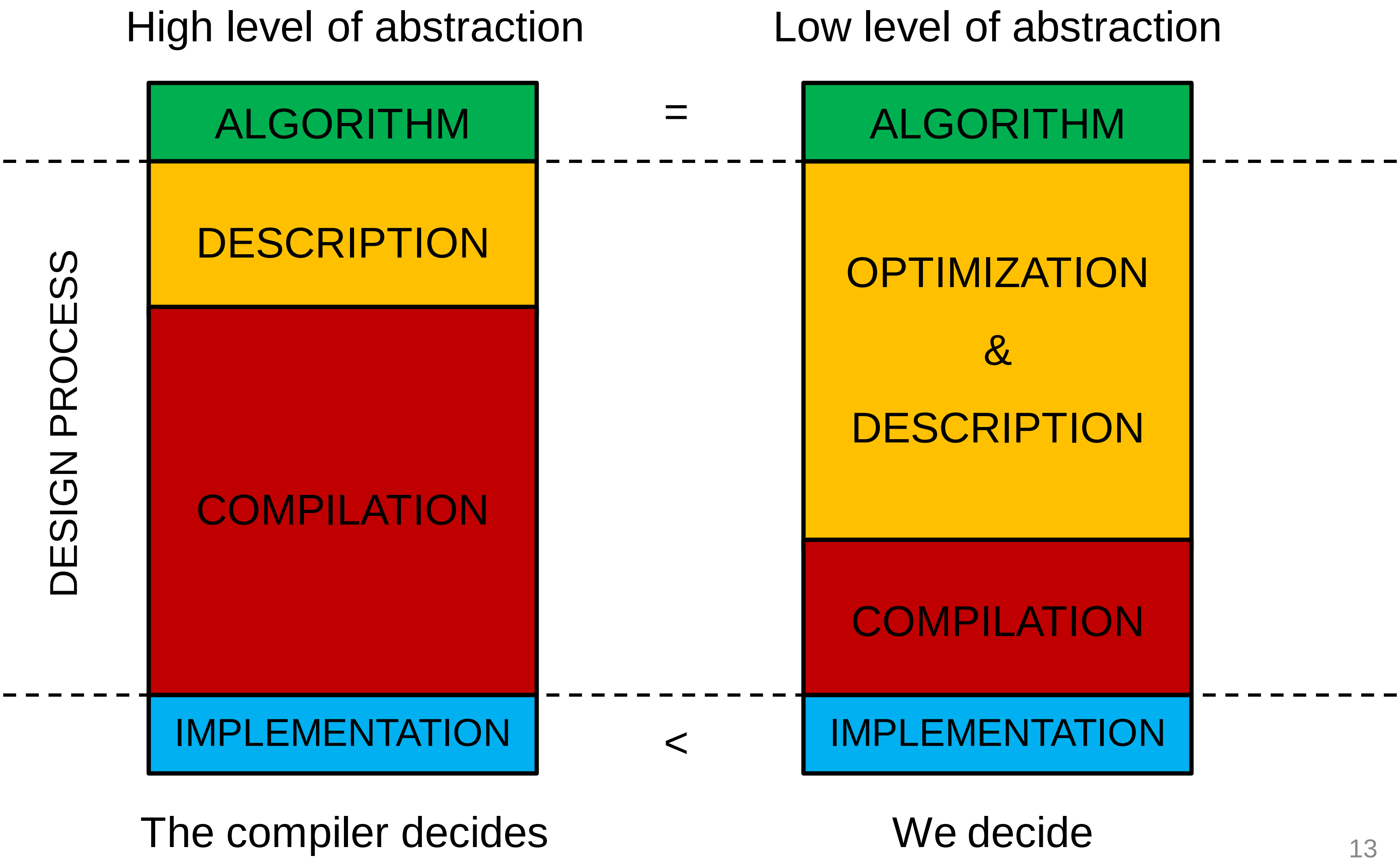


WE

5x

Why?

LEVEL OF ABSTRACTION



High level of abstraction

Low level of abstraction

DESIGN PROCESS

ALGORITHM

DESCRIPTION

COMPILATION

IMPLEMENTATION

=

<

ALGORITHM

OPTIMIZATION

&

DESCRIPTION

COMPILATION

IMPLEMENTATION

The compiler decides

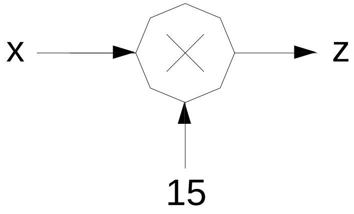
We decide

ABSTRACTION vs PERFORMANCE

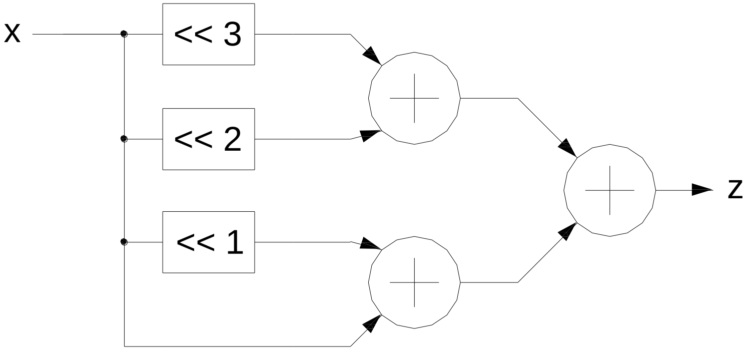
LANGUAGE DESCRIPTION

HARDWARE IMPLEMENTATION

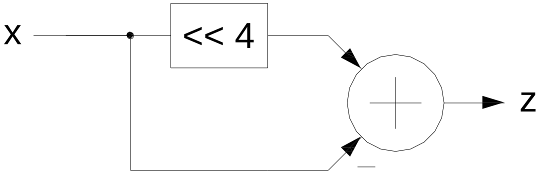
z = 15 * x;



z = (x<<3)+(x<<2)+(x<<1) +x;



z = (x<<4)-x;



UNDERSTANDING GPUs

1.- The performance is related to the computation time. The lower the computation time, the higher the performance. Try to simplify the operations in the algorithm.

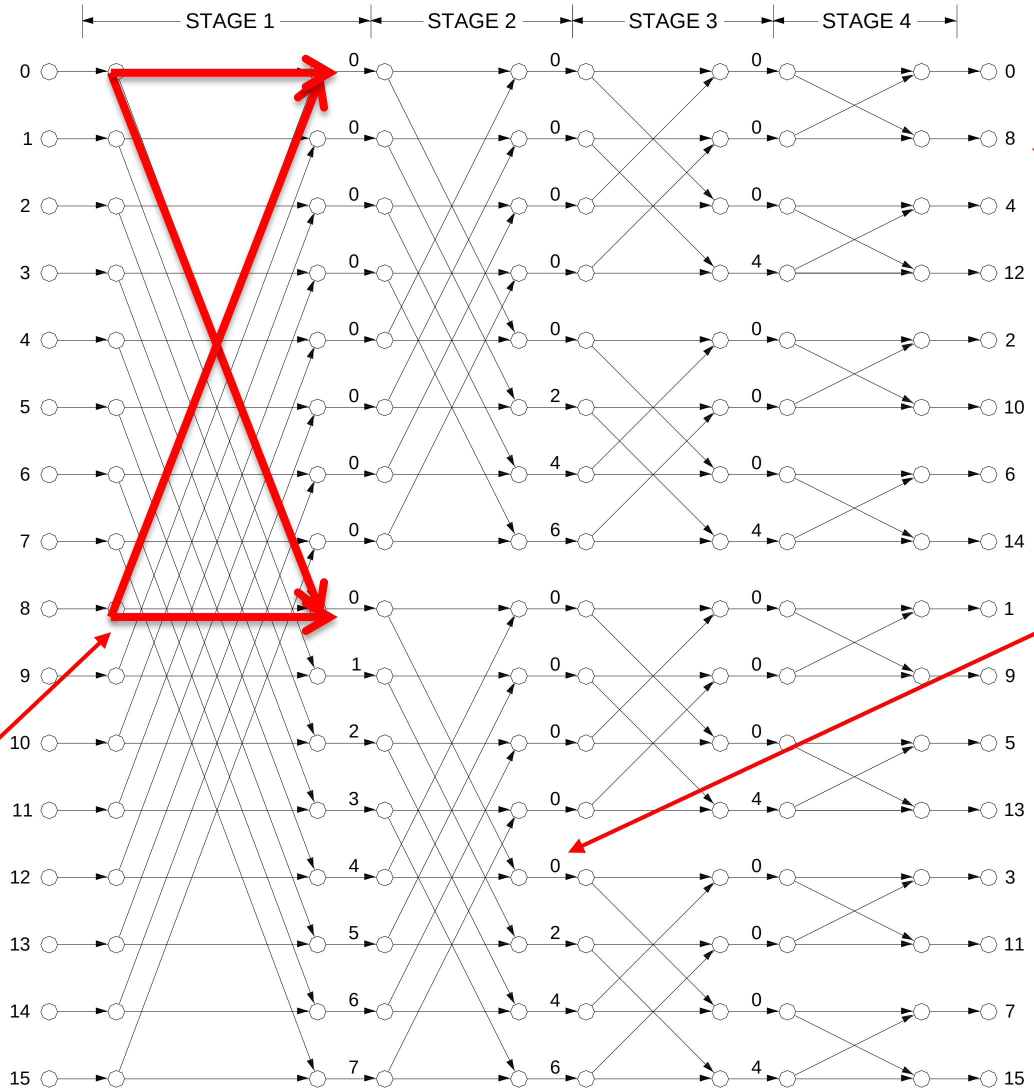
2.- Transactions to global memory very expensive. Try to avoid or minimize. Try to use shared memory.

3.- Threads must be synchronized if we want to share information among them. Unless they are in the same warp. Try to reduce the number of synchronization points.

4.- We have to calculate the index of the data processed by each thread. Try to minimize the number of index calculations.

5.- Threads process data in parallel and the synchronization is not possible until all the threads have finished the calculations. Balance the load among thread.

FFT FLOW GRAPH (RADIX -2)



x[n]

n

N points

$\log_r N$ stages

butterflies

(radix-2)

X[k]

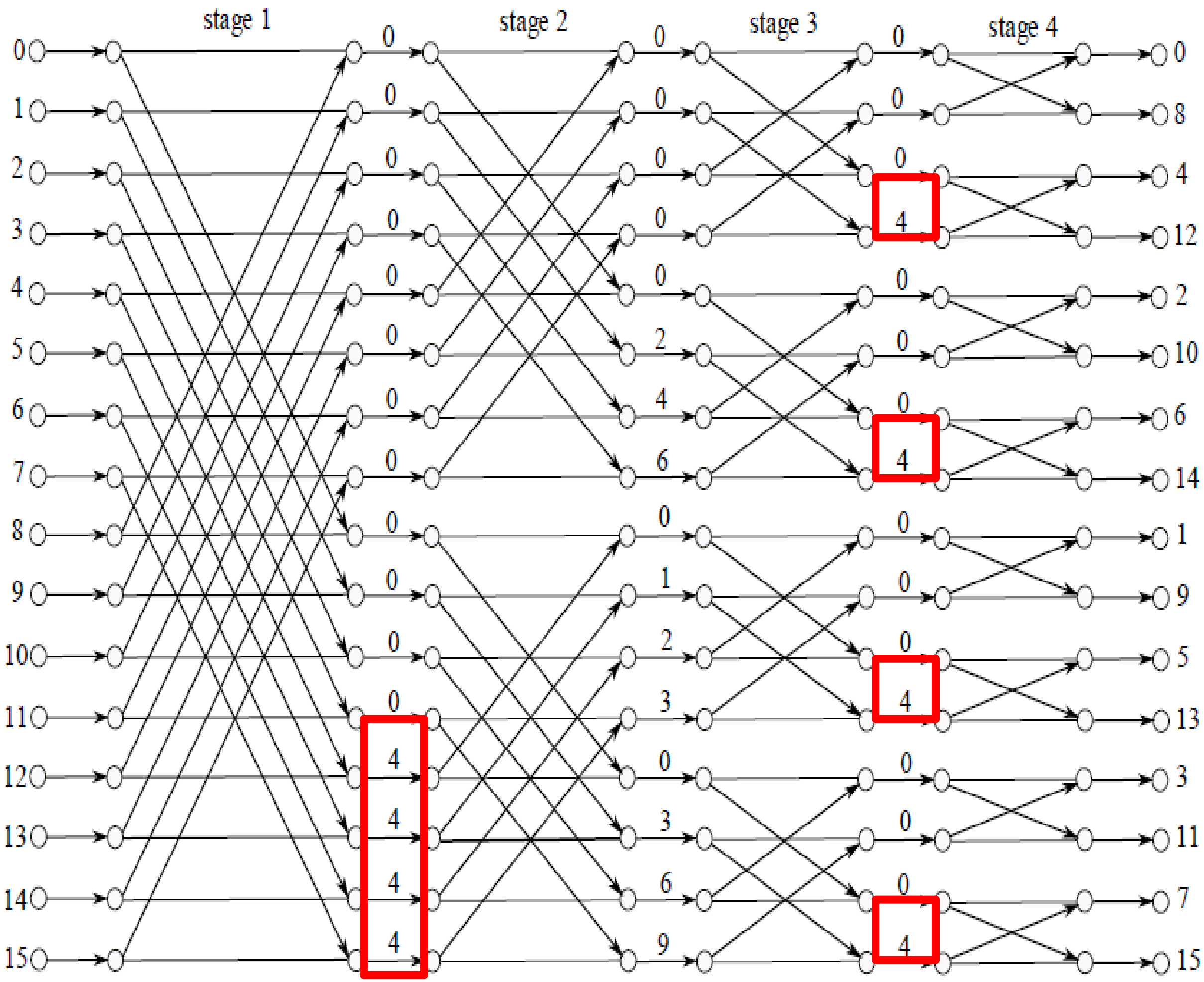
k

rotations

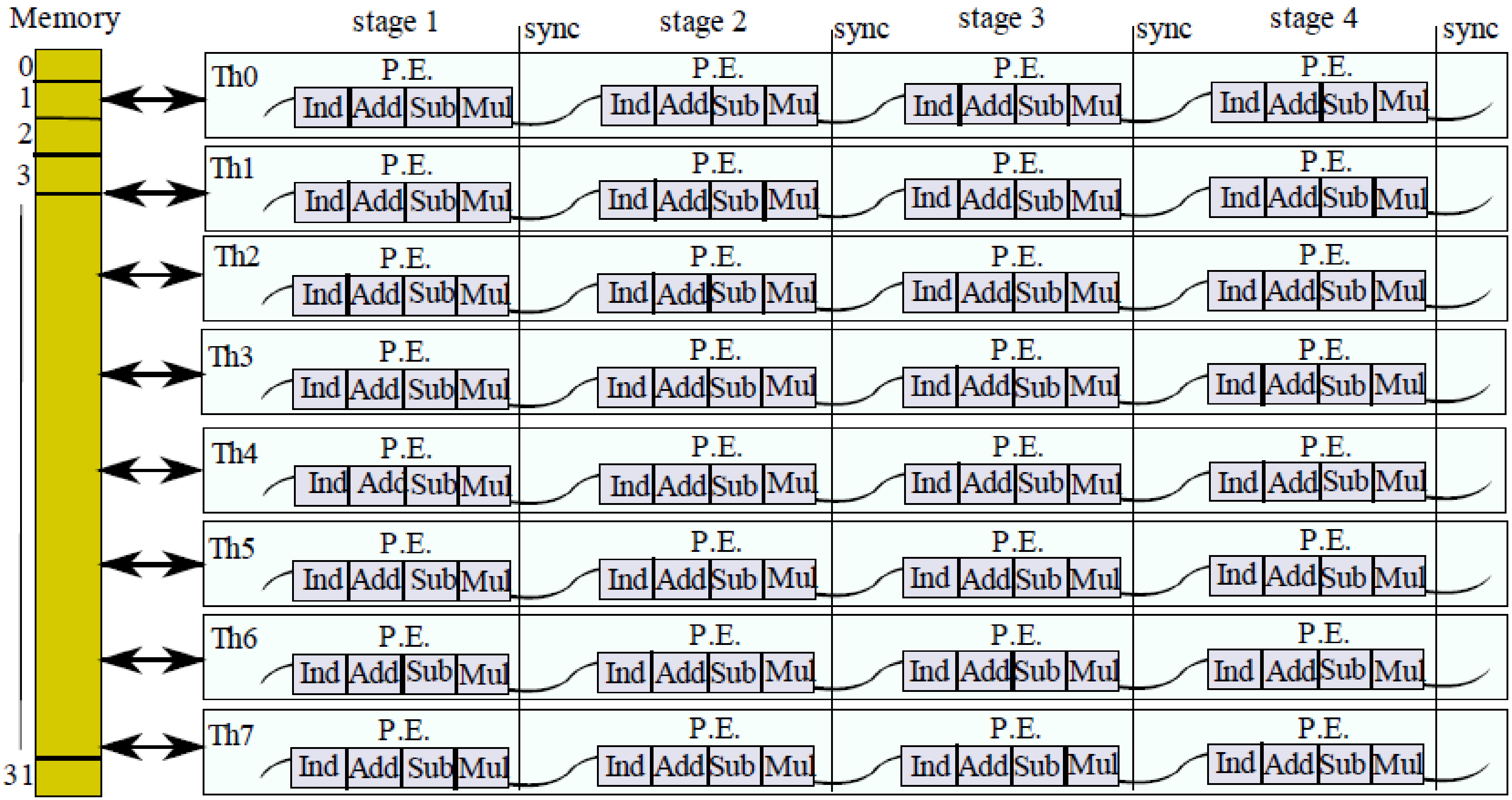
$$e^{-j\frac{2\pi}{N}\phi}$$

1. SIMPIFY THE ALGORITHM

USE RADIX-2^2

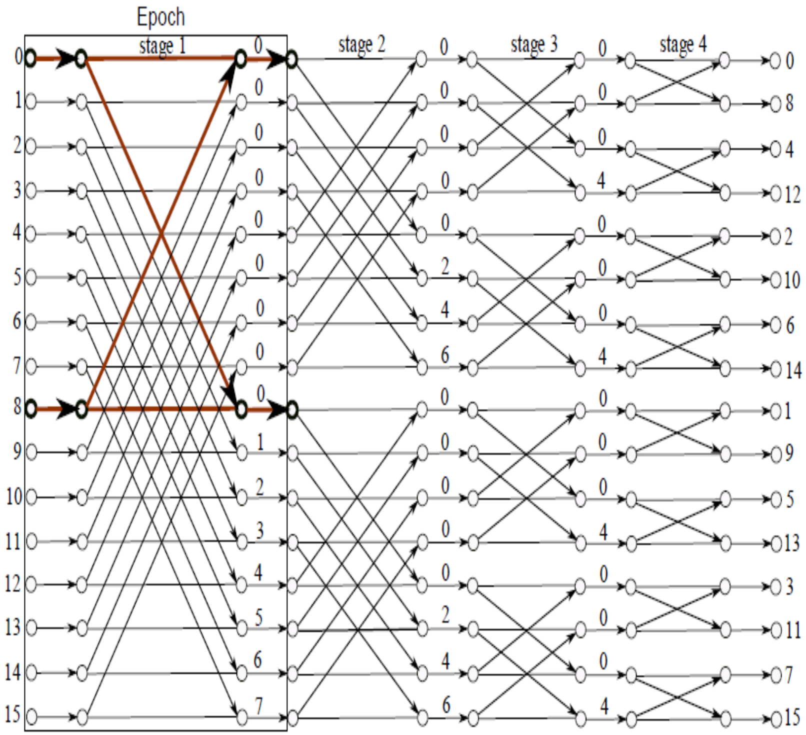


2. USE SHARED MEMORY

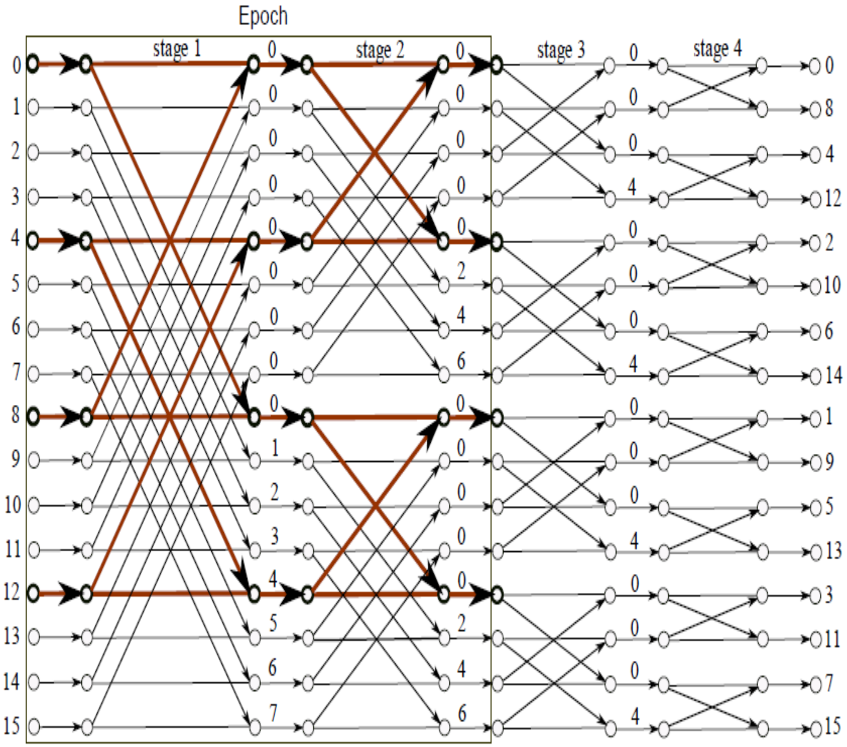


3. REDUCE SYNC. POINTS

USE WORD GROUPS



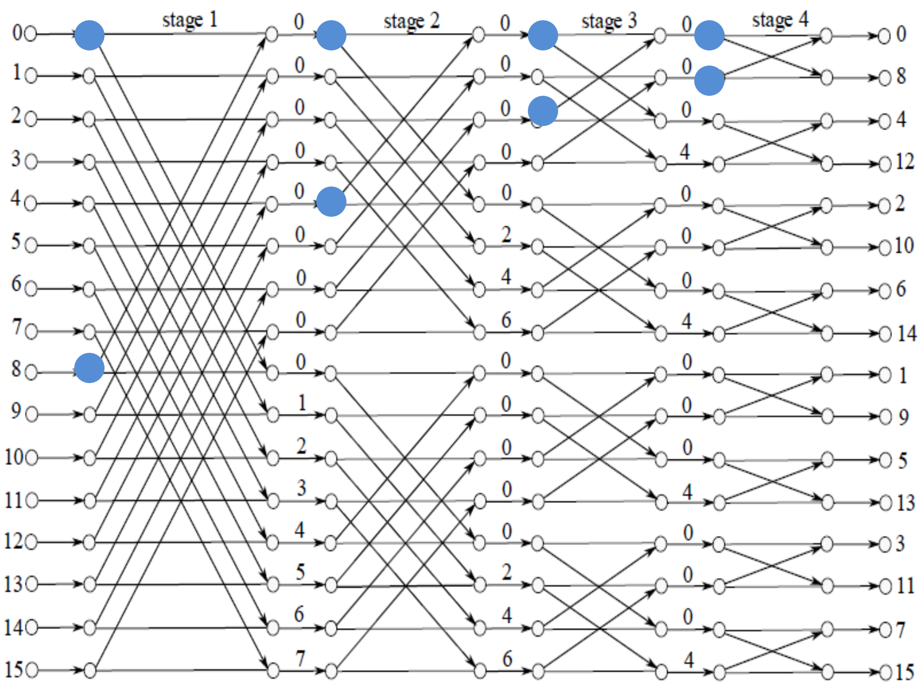
2-word group



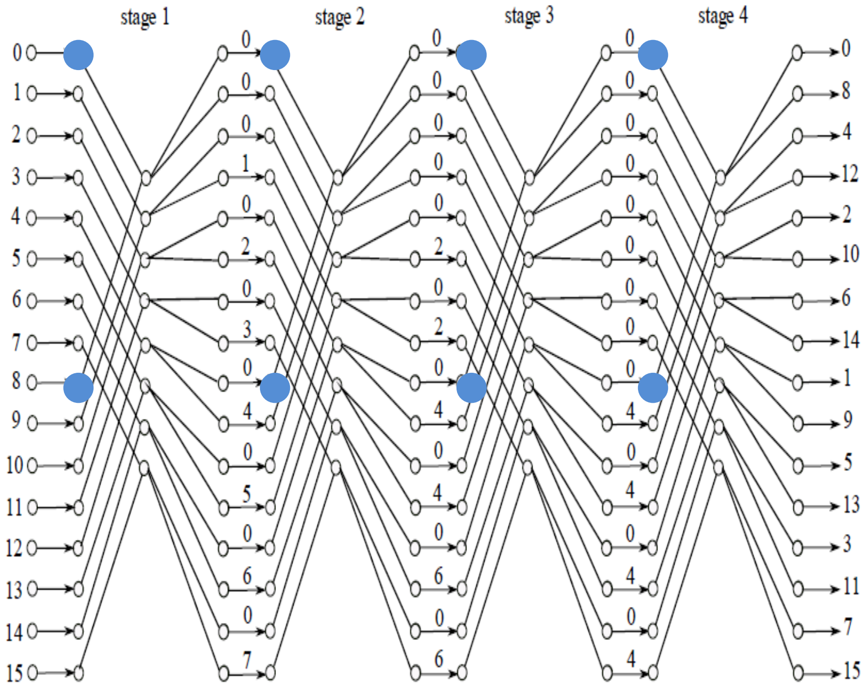
4-word group

4. REDUCE INDEX CALCULATIONS

USE CONSTANT GEOMETRY



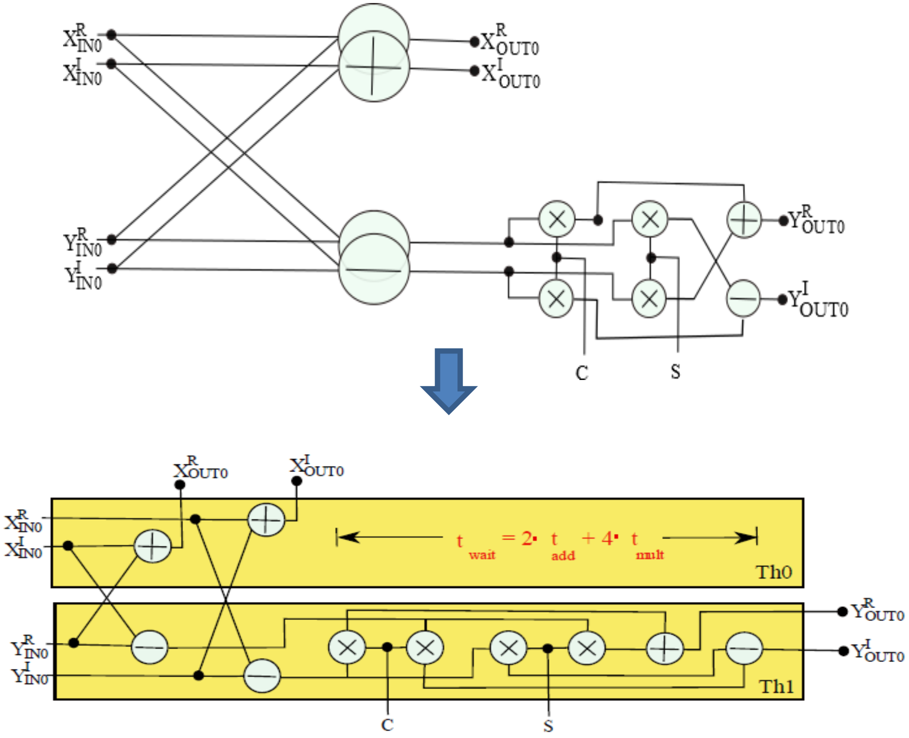
Conventional flow graph



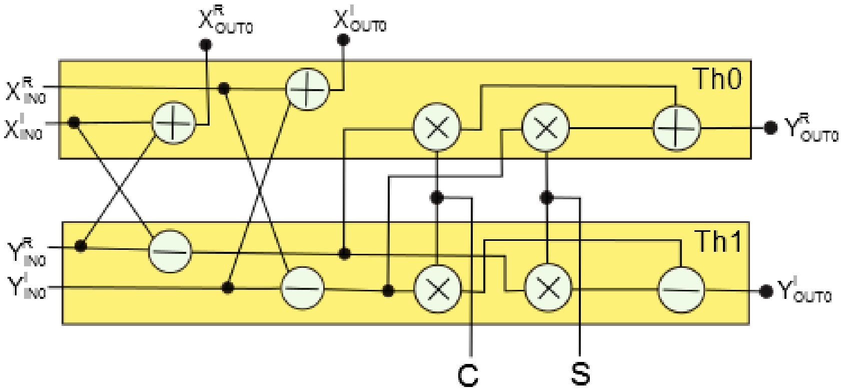
Constant Geometry

5. BALANCE LOAD AMONG THREADS

USE SCHEDULING



Unbalanced scheduling



Balanced scheduling

CONCLUSIONS

- Optimization:
  - Depends on the details and the level of abstraction.
  - Requires to understand in-depth what you are doing.

- Teamwork makes a difference.

- GPUs are fun.

THE END

TO BE CONTINUED…